WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Thursday, April 29, 2004

Hide?	Set Name	Query	Hit Count
		DB=USPT; PLUR=NO; OP=ADJ	
☐	L12	US-6453407-B1.did.	1
		DB=DWPI; PLUR=NO; OP=ADJ	
☐	L11	(yanni and shenderovitch).in.	2
		DB=PGPB,USPT; PLUR=NO; OP=ADJ	
☐	L10	treat.xp. and bops.as.	3
☐	L9	treat.xa. and siw	0
☐	L8	treat.xa. and bops.asn.	0
☐	L7	treat.xa. and bops.as.	0
☐	L6	247686.ap.	5
☐	L5	vliw with (coprocessor$1 or co-processor$1)	19
☐	L4	automated processor generation system	5
☐	L3	506502.ap.	2
☐	L2	246047.ap.	4
☐	L1	6282633.pn.	1

END OF SEARCH HISTORY

First Hit Fwd Refs

End of Result Set

Generate Collection Print

L59: Entry 2 of 2 File: USPT Aug 4, 1998

US-PAT-NO: 5790881
DOCUMENT-IDENTIFIER: US 5790881 A

TITLE: Computer system including coprocessor devices simulating memory interfaces

DATE-ISSUED: August 4, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Nguyen; Julien T.	Redwood City	CA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Sigma Designs, Inc.	Fremont	CA			02

APPL-NO: 08/ 385249 [PALM]
DATE FILED: February 7, 1995

INT-CL: [06] G06 F 9/44

US-CL-ISSUED: 395/800.34; 395/500, 395/503, 395/527
US-CL-CURRENT: 712/34; 345/503, 703/24, 703/27

FIELD-OF-SEARCH: 395/162, 395/163, 395/134, 395/135, 395/800.34, 395/500, 395/503, 395/527

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	3846762	November 1974	Gregory et al.	341/126
☐	3947826	March 1976	Bockwoldt	348/441
☐	4353057	October 1982	Bernet et al.	341/136
☐	4394650	July 1983	Long et al.	45/23
☐	4425581	January 1984	Schweppe et al.	348/570
☐	4528561	July 1985	Kitamura	345/195
☐	4587633	May 1986	Wang et al.	395/200.64

☐	4628479	December 1986	Borg et al.	395/479
☐	4684936	August 1987	Brown et al.	345/116
☐	4757310	July 1988	Katsura et al.	345/28
☐	4779210	October 1988	Katsura et al.	395/134
☐	4811205	March 1989	Normington et al.	395/502
☐	4862392	August 1989	Steiner	395/127
☐	4870406	September 1989	Gupta et al.	340/70
☐	4876600	October 1989	Pietzsch et al.	348/588
☐	4905189	February 1990	Brunolli	395/501
☐	4916301	April 1990	Mansfield et al.	345/133
☐	4947257	August 1990	Fernandez et al.	345/585
☐	4947342	August 1990	Katsura et al.	395/136
☐	4953101	August 1990	Kellcher et al.	395/505
☐	4994912	February 1991	Lumelsky et al.	348/441
☐	5046023	September 1991	Katsura et al.	395/134
☐	5065346	November 1991	Kawai et al.	395/128
☐	5097257	March 1992	Clough et al.	345/132
☐	5111292	May 1992	Kuriacose et al.	348/384
☐	5111409	May 1992	Gasper et al.	395/807
☐	5122875	June 1992	Raychaudhuri et al.	348/390
☐	5138307	August 1992	Tatsumi	345/121
☐	5151875	September 1992	Sato	364/784.03
☐	5157716	October 1992	Naddor et al.	379/92.01
☐	5168356	December 1992	Acampora et al.	348/409
☐	5191410	March 1993	McCalley et al.	348/14
☐	5191548	March 1993	Balkanski et al.	364/725.03
☐	5196946	March 1993	Balkanski et al.	358/433
☐	5208745	May 1993	Quentin et al.	364/188
☐	5220312	June 1993	Lumelsky et al.	345/190
☐	5231492	July 1993	Dangi et al.	348/17
☐	5253078	October 1993	Balkanski et al.	358/426
☐	5270832	December 1993	Balkanski et al.	358/432
☐	5289276	February 1994	Siracusa et al.	348/469
☐	5309567	May 1994	Mizukami	395/290
☐	5329630	July 1994	Baldwin	395/497.64
☐	5333261	July 1994	Guttal et al.	395/501
☐	5341318	August 1994	Balkanski et al.	364/725.03
	5371861	December 1994	Keener et al.	395/309

☐				
☐	5379356	January 1995	Purcell et al.	382/233
☐	5392239	February 1995	Margulis et al.	365/189.01
☐	5397853	March 1995	Koguchi	434/307A
☐	5402147	March 1995	Chen et al.	345/115
☐	5406306	April 1995	Siann et al.	345/115
☐	5416749	May 1995	Lai	365/240
☐	5426756	June 1995	Shyi et al.	395/486
☐	5434913	July 1995	Tung et al.	379/202
☐	5446501	August 1995	Takemoto et al.	348/620
☐	5450542	September 1995	Lchman	395/512
☐	5450544	September 1995	Dixon et al.	395/507
☐	5471576	November 1995	Yee	395/807
☐	5526503	June 1996	Kim	395/413

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
0 384 257	August 1990	EP	
0 384 419	August 1990	EP	

OTHER PUBLICATIONS

Brunhoff, Todd; "VEX Provides Mechanism for Integrating Graphics and Video"; Computer Technology Review; 10 Nov. 1990; No. 1, pp. 107-111.

ART-UNIT: 232

PRIMARY-EXAMINER: Harrity; John E.

ASSISTANT-EXAMINER: Follansbee; John

ATTY-AGENT-FIRM: The Law Offices of Steven A. Swernofsky

ABSTRACT:

A method and system for coupling a coprocessor to a master device, in which the coprocessor emulates an memory interface to the master device, like that of a memory device. The coprocessor is coupled to a memory bus and receives memory accesses directed to a set of addresses not covered by memory devices also coupled to the memory bus. The coprocessor is disposed to receive data written from the master device, perform a coprocessing function on that data, and respond to a read data command from the master device with processing results. The coprocessor uses memory block transfers to read data from and write data to memory devices also coupled to the memory bus. A general purpose computer system comprises a central processor and memory coupled to a PCI bus, a graphics processor and graphics memory coupled to the PCI bus, and a coprocessor coupled to the graphics processor and graphics memory. The coprocessor is adapted to compute, in response to data written

to it by the graphics processor, a graphical function such as a 3D processing function, MPEG video compression or decompression, a raytracing function, or some related function in support of graphics processing. The coprocessor may communicate with the central processor and its memory using a memory access operation performed by the central processor, and may communicate with the graphics memory using a memory block transfer performed by the graphics processor. The coprocessor may emulate a memory wait condition or memory wait state using read-ready and write-ready flags which are readable by software executing on the general processor.

18 Claims, 4 Drawing figures

First Hit Fwd Refs

Generate Collection Print

L59: Entry 1 of 2 File: USPT Nov 10, 1998

US-PAT-NO: 5835750
DOCUMENT-IDENTIFIER: US 5835750 A
** See image for Certificate of Correction **

TITLE: User transparent system using any one of a family of processors in a single socket

DATE-ISSUED: November 10, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Pan-Ratzlaff; Ruby Y.	Austin	TX		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Dell USA, L.P.	Round Rock	TX			02

APPL-NO: 08/ 744671 [PALM]
DATE FILED: November 6, 1996

PARENT-CASE:
This is a Continuation of application Ser. No. 08/538,676, filed Oct. 2, 1995, now abandoned, which is a continuation of application Ser. No. 07/766,877, filed Sep. 27, 1991, now abandoned.

INT-CL: [06] G06 F 9/455

US-CL-ISSUED: 395/500; 395/800.34, 395/800.37, 395/830, 395/836, 395/527
US-CL-CURRENT: 712/37; 703/27, 710/10, 710/16, 712/34, 713/1

FIELD-OF-SEARCH: 395/820, 395/520, 395/500, 395/830, 395/836, 395/820.34, 395/820.37, 395/527

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	4964074	October 1990	Suzuki et al.	364/927.81
☐	4972470	November 1990	Farago	380/3
☐	5109503	April 1992	Cruickshank et al.	395/500

☐	5134713	July 1992	Miller et al.	395/800
☐	5321827	June 1994	Lu et al.	395/500
☐	5546563	August 1996	Chuang	395/500

ART-UNIT: 273

PRIMARY-EXAMINER: Bowler; Alyssa H.

ASSISTANT-EXAMINER: Follansbe; John T.

ATTY-AGENT-FIRM: Skjerven, Morrill, MacPherson, Franklin & Friel, L.L.P. Terrile; Stephen A.

ABSTRACT:

A digital computer system having a socket capable of accepting any one of a family of processors, the family being defined as those processors having commonality of their respective basic input/output system code. Each processor has assigned pins for conducting specified signals, the pins engaging the socket. There is dissimilarity between at least two of the processors of correspondence where at least one of the specified signals, of one of the processors, is assigned to a different pin from the other processors. When such dissimilarity is present, the signal is redirected to the appropriate designated pin for the particular type of processor.

24 Claims, 6 Drawing figures

First Hit Fwd Refs

End of Result Set

☐ Generate Collection Print

L1: Entry 1 of 1 File: USPT Aug 28, 2001

US-PAT-NO: 6282633
DOCUMENT-IDENTIFIER: US 6282633 B1

TITLE: High data density RISC processor

DATE-ISSUED: August 28, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Killian; Earl A.	Los Altos Hills	CA		
Gonzalez; Ricardo E.	Menlo Park	CA		
Dixit; Ashish B.	Mountain View	CA		
Lam; Monica	Menlo Park	CA		
Lichtenstein; Walter D.	Belmont	MA		
Rowen; Christopher	Santa Cruz	CA		
Ruttenberg; John C.	Newton	MA		
Wilson; Robert P.	Palo Alto	CA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Tensilica, Inc.	Santa Clara	CA			02

APPL-NO: 09/ 192395 [PALM]
DATE FILED: November 13, 1998

INT-CL: [07] G06 K 9/30

US-CL-ISSUED: 712/208; 712/210, 712/226
US-CL-CURRENT: 712/208; 712/210, 712/226

FIELD-OF-SEARCH: 712/24, 712/41, 712/200, 712/208, 712/210, 712/223, 712/225, 712/226

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	5155820	October 1992	Gibson	712/210
	5426743	June 1995	Phillips et al.	712/221

☐				
☐	5581717	December 1996	Boggs et al.	712/208
☐	5638524	June 1997	Kiuchi et al.	712/221
☐	5991870	November 1999	Koumura et al.	712/208

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
0 363 222 A2	April 1990	EP	
0 374 419 A2	June 1990	EP	
0 660 223 A2	June 1995	EP	
0 696 772 A2	February 1996	EP	
0 924 601 A2	June 1999	EP	
WO 93/01543	January 1993	WO	

OTHER PUBLICATIONS

Texas Instruments, "TMS32010 User's Guide", 1983, p. 3-7.*
Lefugy, C. et al., "improving Code Density using Compression Techniques", 12/97. p. 194-204.*
Conte, T.M. et al., "Instruction Fetch Mechanisms for VLIW Architectures Compressed Encodings", 12/96. p. 201-211.

ART-UNIT: 282

PRIMARY-EXAMINER: Niebling; John F.

ASSISTANT-EXAMINER: Whitmore; Stacy

ATTY-AGENT-FIRM: Pillsbury Winthrop LLP

ABSTRACT:

A RISC processor implements an instruction set which, in addition to optimizing a relationship between the number of instructions required for execution of a program, clock period and average number of clocks per instruction, also is designed to optimize the equation S=IS * BI, where S is the size of program instructions in bits, IS is the static number of instructions required to represent the program (not the number required by an execution) and BI is the average number of bits per instruction. Compared to conventional RISC architectures, this processor lowers both BI and IS with minimal increases in clock period and average number of clocks per instruction. The processor provides good code density in a fixed-length high-performance encoding based on RISC principles, including a general register with load/store architecture. Further, the processor implements a simple variable-length encoding that maintains high performance.

21 Claims, 5 Drawing figures

First Hit Fwd Refs

Generate Collection Print

L2: Entry 2 of 4 File: USPT Nov 5, 2002

US-PAT-NO: 6477683
DOCUMENT-IDENTIFIER: US 6477683 B1

TITLE: Automated processor generation system for designing a configurable processor and method for the same

DATE-ISSUED: November 5, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Killian; Earl A.	Los Altos Hills	CA		
Gonzalez; Ricardo E.	Menlo Park	CA		
Dixit; Ashish B.	Mountain View	CA		
Lam; Monica	Menlo Park	CA		
Lichtenstein; Walter D.	Belmont	MA		
Rowen; Christopher	Santa Cruz	CA		
Ruttenberg; John C.	Newton	MA		
Wilson; Robert P.	Palo Alto	CA		
Wang; Albert Ren-Rui	Fremont	CA		
Maydan; Dror Eliezer	Palo Alto	CA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Tensilica, Inc.	Santa Clara	CA			02

APPL-NO: 09/ 246047 [PALM]
DATE FILED: February 5, 1999

INT-CL: [07] G06 F 17/50

US-CL-ISSUED: 716/1; 716/18
US-CL-CURRENT: 716/1; 716/18

FIELD-OF-SEARCH: 716/1-21, 712/32, 712/36, 712/37, 712/41, 712/23, 712/15, 712/1

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	5450586	September 1995	Kuzara et al.	717/4

☐	5535331	July 1996	Swoboda et al.	714/45
☐	5748875	May 1998	Tzori	714/29
☐	5819050	October 1998	Boehling et al.	710/104
☐	5854929	December 1998	Van Praet et al.	717/5
☐	5870588	February 1999	Rompaey et al.	703/13
☐	5999734	December 1999	Willis et al.	717/6
☐	6006022	December 1999	Rhim et al.	716/1
☐	6182206	January 2001	Baxter	712/43
☐	6195593	February 2001	Nguyen	700/97

OTHER PUBLICATIONS

Hartoog et al, "Generaion of Sotware Tools from Processor Descriptions for Hardware/Software Codesign," ACM, Jun. 1997, pp. 303-306.*
Freericks "The nML Machine Description Formalism" (Bericht 1991/15 pp. 3-41).
Fauth et al. "Describing Instruction Set Processors Using nML" (Proc. Euro. Design & Test Conf., Paris, Mar. 1995, IEEE 1995, 5 pp.).
Internet Publication http://www.retarget.com/brfchschk.html (19 pp) No date.
Internet Publication http://www.synopsys.com/products/designware/8051_ds.html (8 pp) No date.
Internet Publication http://www.synopsys.com/products/designware/dwpci_ds.html (16 pp) No date.
Internet Publication http://www.lexra.com/product.html (11 pp) No date.
Internet Publication http://www.risccores.com/html/body_aboutarc.htm (13 pp) No date.
Reference Manual, Tensilica "Xtensa" Instruction Set Architecture (ISA) Reference Manual, Revision 1.0, Tensilica, Inc. No date.

ART-UNIT: 2825

PRIMARY-EXAMINER: Siek; Vuthe

ATTY-AGENT-FIRM: Pillsbury Winthrop LLP

ABSTRACT:

An automated processor design tool uses a description of customized processor instruction set extensions in a standardized language to develop a configurable definition of a target instruction set, a Hardware Description Language description of circuitry necessary to implement the instruction set, and development tools such as a compiler, assembler, debugger and simulator which can be used to develop applications for the processor and to verify it. Implementation of the processor circuitry can be optimized for various criteria such as area, power consumption, speed and the like. Once a processor configuration is developed, it can be tested and inputs to the system modified to iteratively optimize the processor implementation. By providing a constrained domain of extensions and optimizations, the process can be automated to a high degree, thereby facilitating fast and reliable development.

104 Claims, 15 Drawing figures

First Hit Fwd Refs

End of Result Set

Generate Collection Print

L4: Entry 5 of 5 File: USPT Nov 5, 2002

US-PAT-NO: 6477683
DOCUMENT-IDENTIFIER: US 6477683 B1

TITLE: Automated processor generation system for designing a configurable processor and method for the same

DATE-ISSUED: November 5, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Killian; Earl A.	Los Altos Hills	CA		
Gonzalez; Ricardo E.	Menlo Park	CA		
Dixit; Ashish B.	Mountain View	CA		
Lam; Monica	Menlo Park	CA		
Lichtenstein; Walter D.	Belmont	MA		
Rowen; Christopher	Santa Cruz	CA		
Ruttenberg; John C.	Newton	MA		
Wilson; Robert P.	Palo Alto	CA		
Wang; Albert Ren-Rui	Fremont	CA		
Maydan; Dror Eliezer	Palo Alto	CA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Tensilica, Inc.	Santa Clara	CA			02

APPL-NO: 09/ 246047 [PALM]
DATE FILED: February 5, 1999

INT-CL: [07] G06 F 17/50

US-CL-ISSUED: 716/1; 716/18
US-CL-CURRENT: 716/1; 716/18

FIELD-OF-SEARCH: 716/1-21, 712/32, 712/36, 712/37, 712/41, 712/23, 712/15, 712/1

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL

☐	5450586	September 1995	Kuzara et al.	717/4
☐	5535331	July 1996	Swoboda et al.	714/45
☐	5748875	May 1998	Tzori	714/29
☐	5819050	October 1998	Boehling et al.	710/104
☐	5854929	December 1998	Van Praet et al.	717/5
☐	5870588	February 1999	Rompaey et al.	703/13
☐	5999734	December 1999	Willis et al.	717/6
☐	6006022	December 1999	Rhim et al.	716/1
☐	6182206	January 2001	Baxter	712/43
☐	6195593	February 2001	Nguyen	700/97

OTHER PUBLICATIONS

Hartoog et al, "Generaion of Sotware Tools from Processor Descriptions for Hardware/Software Codesign," ACM, Jun. 1997, pp. 303-306.*
Freericks "The nML Machine Description Formalism" (Bericht 1991/15 pp. 3-41).
Fauth et al. "Describing Instruction Set Processors Using nML" (Proc. Euro. Design & Test Conf., Paris, Mar. 1995, IEEE 1995, 5 pp.).
Internet Publication http://www.retarget.com/brfchschk.html (19 pp) No date.
Internet Publication http://www.synopsys.com/products/designware/8051_ds.html (8 pp) No date.
Internet Publication http://www.synopsys.com/products/designware/dwpci_ds.html (16 pp) No date.
Internet Publication http://www.lexra.com/product.html (11 pp) No date.
Internet Publication http://www.risccores.com/html/body_aboutarc.htm (13 pp) No date.
Reference Manual, Tensilica "Xtensa" Instruction Set Architecture (ISA) Reference Manual, Revision 1.0, Tensilica, Inc. No date.

ART-UNIT: 2825

PRIMARY-EXAMINER: Siek; Vuthe

ATTY-AGENT-FIRM: Pillsbury Winthrop LLP

ABSTRACT:

An automated processor design tool uses a description of customized processor instruction set extensions in a standardized language to develop a configurable definition of a target instruction set, a Hardware Description Language description of circuitry necessary to implement the instruction set, and development tools such as a compiler, assembler, debugger and simulator which can be used to develop applications for the processor and to verify it. Implementation of the processor circuitry can be optimized for various criteria such as area, power consumption, speed and the like. Once a processor configuration is developed, it can be tested and inputs to the system modified to iteratively optimize the processor implementation. By providing a constrained domain of extensions and optimizations, the process can be automated to a high degree, thereby facilitating fast and reliable development.

104 Claims, 15 Drawing figures

First Hit

Generate Collection Print

L5: Entry 4 of 19 File: PGPB Jun 27, 2002

PGPUB-DOCUMENT-NUMBER: 20020083306
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20020083306 A1

TITLE: Digital signal processing apparatus

PUBLICATION-DATE: June 27, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Pessolano, Francesco	Eindhoven		NL	
Kessels, Jozef Laurentius Wilhelmus	Eindhoven		NL	
Peeters, Adrianus Marinus Gerardus	Eindhoven		NL	

APPL-NO: 10/ 020019 [PALM]
DATE FILED: December 7, 2001

FOREIGN-APPL-PRIORITY-DATA:

COUNTRY	APPL-NO	DOC-ID	APPL-DATE
EP	00310905.5	2000EP-00310905.5	December 7, 2000

INT-CL: [07] G06 F 9/00

US-CL-PUBLISHED: 712/220
US-CL-CURRENT: 712/220

REPRESENTATIVE-FIGURES: 5

ABSTRACT:

The present invention relates to a digital signal processing apparatus for executing a plurality of operations, comprising a plurality of functional units (10) wherein each functional unit (10) is adapted to execute operations, and control means for controlling said functional units (10), wherein said control means comprises a plurality of control units (12) wherein at least one control unit (12) is operatively associated to any functional unit (10), respectively, for controlling its function, and each functional unit (10) is adapted to execute operations in an autonomous manner under control by the control unit (12) associated thereto, and/or wherein provided is a FIFO (first-in/fist-out) register means (14) adapted for supporting data-flow communication among said functional units (10). Further the present invention relates to a method for processing digital signals in digital signal processing apparatus comprising a plurality of functional units (10) wherein each functional unit (10) is adapted execute operations, and wherein said functional units (10) are controlled by a plurality of control units (12) wherein at least one control unit (12) is operatively associated to any functional unit (10), respectively, so that each functional unit (10) is

able to execute operations in an autonomous manner under control by the control unit (12) associated thereto, and/or wherein data-flow communication among said functional units (10) is supported by FIFO (first-in/first-out) register means (14).

First Hit

Generate Collection Print

L5: Entry 4 of 19 File: PGPB Jun 27, 2002

DOCUMENT-IDENTIFIER: US 20020083306 A1
TITLE: Digital signal processing apparatus

Summary of Invention Paragraph:
[0002] Such an apparatus and a method are usually implemented in digital signal processors (DSPs). To increase their performance, the digital signal processors contain several processing units which normally operate in small loops. Two conventional solutions exist, namely the provision of (1.) VLIW processors comprising several functional units and a central control, and (2.) a control processor with co-processors each of which performs a fixed function autonomously.

Summary of Invention Paragraph:
[0013] It is an object of the present invention to still further increase the performance and in particular to obtain a digital signal processing apparatus and method which combine the flexibility of a VLIW processor with the coarse grain parallelism offered by the provision of co-processors.

Summary of Invention Paragraph:
[0020] After all, the present invention provides a solution which combines the flexibility of VLIW processors with the coarse grain parallelism offered by co-processors.

First Hit Fwd Refs

Generate Collection Print

L5: Entry 8 of 19 File: USPT Jan 20, 2004

US-PAT-NO: 6681052
DOCUMENT-IDENTIFIER: US 6681052 B2

TITLE: Methods and systems for performing inverse quantization and inverse weighting of DV video

DATE-ISSUED: January 20, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Luna; Amelia Carino	San Jose	CA		
Wang; Jason Naxin	San Jose	CA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Sony Corporation	Tokyo			JP	03
Sony Electronics, Inc.	Park Ridge	NJ			02

APPL-NO: 09/ 764320 [PALM]
DATE FILED: January 16, 2001

PARENT-CASE:
This application claims the benefit of U.S. Provisional Application No. 60,176,257, filed Jan. 15, 2000.

INT-CL: [07] G06 K 9/36, H04 B 1/66

US-CL-ISSUED: 382/250; 382/251, 375/240.24, 375/240.2
US-CL-CURRENT: 382/250; 375/240.2, 375/240.24, 382/251

FIELD-OF-SEARCH: 382/173, 382/232, 382/233, 382/248, 382/250, 382/251, 375/240.03, 375/240.2, 375/240.18, 375/240.22, 375/240.24

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	5455629	October 1995	Sun et al.	375/240.27
☐	5724097	March 1998	Hibi et al.	375/240.04
☐	6172621	January 2001	Iwata	341/50

☐	6348945	February 2002	Hayakawa	375/240.18
☐	6389171	May 2002	Washington	382/233
☐	6507614	January 2003	Li	375/240.03

ART-UNIT: 2621

PRIMARY-EXAMINER: Johns; Andrew W.

ASSISTANT-EXAMINER: Alavi; Amir

ATTY-AGENT-FIRM: Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P.

ABSTRACT:

In methods and systems consistent with the present invention, the process of inverse quantization is performed by determining class number and quantization number for each block of received quantized DCT coefficients, determining a first shift value based on the class number and quantization number and a second shift value based on the class number and a combination type, and shifting the entire block of DCT coefficients based on the first and second shift values. Alternatively, the inverse quantization may be combined with inverse weighting step by pre-shifting a set of weighting tables, one for each area number combination. A pre-shifted weighting matrix is then selected based on the second shift value and multiplied by the shifted matrix of DCT coefficients. In another embodiment, a pre-shifted weighting table is selected based on the class number and combination type and then multiplied by the shifted matrix of DCT coefficients.

17 Claims, 13 Drawing figures

L5: Entry 8 of 19 File: USPT Jan 20, 2004

DOCUMENT-IDENTIFIER: US 6681052 B2
TITLE: Methods and systems for performing inverse quantization and inverse weighting of DV video

Detailed Description Text (28):
As shown in FIG. 11, media processor 1100 comprises a co-processor 1111, VLIW Core Processor 1120, memory controller 1130, data streamer 1140, I/O interface 1150, and PCI unit 1160, operatively connected by an internal bus 1165. Media processor 1100 may also be operatively connected to an external memory 1170.

Detailed Description Text (29):
Co-processor 1111 comprises one or more co-processors that execute in parallel with VLIW Core Processor 1120. Co-processor 1111 may comprise, for example, a Variable Length Encoder/Decoder ("VLx") processor, such as a 16-bit RISC co-processor with multiple 16-bit registers that offload VLIW CPU 522 from bit sequential tasks of variable length encoding and decoding. Co-processor 1111 may also comprise, for example, a video scalar co-processor.

Detailed Description Text (33):
I/O Interface 1150 transforms the decoded data into analog output data in, for example, NTSC format. Peripheral Component Interconnect ("PCI") unit 1160 allows VLIW interface 1120, Data Streamer 1140, and Co-processor 1111 to initiate bus requests. PCI is a 64-bit bus, though it may be implemented as a 32-bit bus. It can run at clock speeds of 33 or 66 MHz.

L5: Entry 15 of 19 File: USPT Feb 15, 2000

DOCUMENT-IDENTIFIER: US 6026478 A
TITLE: Split embedded DRAM processor

Drawing Description Text (11):
FIG. 8 illustrates an embodiment of the a split VLIW embedded DRAM coprocessor designed in accordance with the present invention.

Detailed Description Text (24):
The VLIW processor 800 and the embedded DRAM coprocessor 810 are operative to jointly execute VLIW programs. That is, the VLIWs read from the cache 805 and the cache 850 form one extended VLIW for the split VLIW processor comprising the VLIW processor 800 and the VLIW extension processor 810. When a program begins, the BPU 825 and the BPU 865 synchronize via the branch interface module 870. The compiler is aware of the extension hardware 810 and treats the embedded DRAM extension processor 810 simply as extra VLIW architectural fields. When the program is compiled, the instructions for the functional units 835 are stored in a VLIW program space serviced by the VLIW program cache 805. The instructions for the functional units 845 are stored in a VLIW extension program space serviced by VLIW cache 850. When a VLIW is fetched from the VLIW program cache 805, a corresponding VLIW extension word is fetched from VLIW program cache 850. To save memory space, the programs in both the VLIW cache 805 and the VLIW extension cache 850 can point to different addresses based on the number of instructions that have been dispatched from the fetched VLIWs. The dispatching of variable numbers of instructions in a VLIW is discussed, for example, in SPRU189B. In the current architecture, the concept is extended to a system that operates in lockstep, but from possibly skewed program addresses. This is readily handled by the compiler and is discussed in greater detail below.

Detailed Description Text (27):
Another aspect of the inventive split VLIW processor architecture is to provide for a fork and join synchronization construct between the BPU 825 and the BPU 865. While application programs execute, it may become advantageous for the VLIW processor 800 and the embedded DRAM coprocessor 810 to fork off separate execution threads. To implement this, the BPU 825 sends program branch synchronization information over the interface 827. Unlike with data dependent branching, the BPU 825 does not instruct the prefetch unit 830 to follow the branch. For a join, the BPU 825 and the BPU 865 both synchronize through the branch interface module 870 by waiting until both BPUs have asserted the join signal. When both BPUs have asserted the join signal, the branch interface module 870 sends a synchronizing signal, and the BPU 825 responds by signaling the prefetch unit 830 to begin prefetching at the join point of the instruction stream, and the BPU 865 similarly signals the prefetch unit 855 to begin prefetching at the join point of the extension instruction stream.

DOCUMENT-IDENTIFIER: US 6026478 A
TITLE: Split embedded DRAM processor

Drawing Description Text (11):
FIG. 8 illustrates an embodiment of the a split VLIW embedded DRAM coprocessor designed in accordance with the present invention.

Detailed Description Text (24):
The VLIW processor 800 and the embedded DRAM coprocessor 810 are operative to jointly execute VLIW programs. That is, the VLIWs read from the cache 805 and the cache 850 form one extended VLIW for the split VLIW processor comprising the VLIW processor 800 and the VLIW extension processor 810. When a program begins, the BPU 825 and the BPU 865 synchronize via the branch interface module 870. The compiler is aware of the extension hardware 810 and treats the embedded DRAM extension processor 810 simply as extra VLIW architectural fields. When the program is compiled, the instructions for the functional units 835 are stored in a VLIW program space serviced by the VLIW program cache 805. The instructions for the functional units 845 are stored in a VLIW extension program space serviced by VLIW cache 850. When a VLIW is fetched from the VLIW program cache 805, a corresponding VLIW extension word is fetched from VLIW program cache 850. To save memory space, the programs in both the VLIW cache 805 and the VLIW extension cache 850 can point to different addresses based on the number of instructions that have been dispatched from the fetched VLIWs. The dispatching of variable numbers of instructions in a VLIW is discussed, for example, in SPRU189B. In the current architecture, the concept is extended to a system that operates in lockstep, but from possibly skewed program addresses. This is readily handled by the compiler and is discussed in greater detail below.

Detailed Description Text (27):
Another aspect of the inventive split VLIW processor architecture is to provide for a fork and join synchronization construct between the BPU 825 and the BPU 865. While application programs execute, it may become advantageous for the VLIW processor 800 and the embedded DRAM coprocessor 810 to fork off separate execution threads. To implement this, the BPU 825 sends program branch synchronization information over the interface 827. Unlike with data dependent branching, the BPU 825 does not instruct the prefetch unit 830 to follow the branch. For a join, the BPU 825 and the BPU 865 both synchronize through the branch interface module 870 by waiting until both BPUs have asserted the join signal. When both BPUs have asserted the join signal, the branch interface module 870 sends a synchronizing signal, and the BPU 825 responds by signaling the prefetch unit 830 to begin prefetching at the join point of the instruction stream, and the BPU 865 similarly signals the prefetch unit 855 to begin prefetching at the join point of the extension instruction stream.

First Hit Fwd Refs

Generate Collection Print

L6: Entry 2 of 5 File: USPT Sep 17, 2002

US-PAT-NO: 6453407
DOCUMENT-IDENTIFIER: US 6453407 B1

TITLE: Configurable long instruction word architecture and instruction set

DATE-ISSUED: September 17, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Lavi; Yoav	Raanana			IL
Rom; Amnon	Azor			IL
Knuth; Robert	Munich			DE
Blum; Rivka	Azor			IL
Yanni; Meny	Azor			IL
Granot; Haim	Azor			IL
Hershko; Anat	Azor			IL
Shenderovitch; Georgiy	Azor			IL
Cohen; Elliot	Raanana			IL
Weingatren; Eran	Tel Hashomer			IL

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Infineon Technologies AG	Munich			DE	03

APPL-NO: 09/ 247686 [PALM]
DATE FILED: February 10, 1999

INT-CL: [07] G06 F 9/22

US-CL-ISSUED: 712/24; 717/5
US-CL-CURRENT: 712/24

FIELD-OF-SEARCH: 712/24, 712/37, 717/4, 717/5

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	4435775	March 1984	Brantingham	711/118

☐	4607332	August 1986	Goldberg	714/8
☐	4610000	September 1986	Lee	365/189
☐	4897813	January 1990	Kumbasar	364/49
☐	4931989	June 1990	Rhodes	712/211
☐	5163139	November 1992	Haigh et al.	712/206
☐	5398321	March 1995	Jeremiah	712/216
☐	5450556	September 1995	Slavenburg et al.	712/235
☐	5634025	May 1997	Breternitz	712/207
☐	5748979	May 1998	Trimberger	712/37
☐	5774737	June 1998	Nakano	712/24
☐	5828897	October 1998	Kirsch	712/43
☐	5859993	January 1999	Snyder	712/208
☐	5950012	September 1999	Shiell	712/209
☐	5966534	October 1999	Cooke	717/5
☐	5983334	November 1999	Coon	712/23
☐	6105109	August 2000	Krumm	711/122
☐	6321322	November 2001	Pechanek	712/24

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
0 723 220	July 1996	EP	

OTHER PUBLICATIONS

"Selecting Predecoded Instructions with a Surrogate", IBM Technical Disclosure Bulletin, XP 000370750.
Joseph A. Fisher: "Very Long Instruction Word Architectures And The ELI-512", Yale University, New Haven, CT, 1983, pp. 140-151.
International Publication WO 97/50030 (Bauer et al.), dated Dec. 31, 1997.
J.A. Barber et al.: "MLID Addressing", IBM Technical Disclosure Bulletin, XP-002069681.

ART-UNIT: 2783

PRIMARY-EXAMINER: Coleman; Eric

ATTY-AGENT-FIRM: Greenberg; Laurence A. Stemer; Werner H. Mayback; Gregory L.

ABSTRACT:

A method for executing instructions in a data processor and improvements to data processor design, which combine the advantages of regular processor architecture and Very Long Instruction Word architecture to increase execution speed and ease of programming, while reducing power consumption. Instructions each consisting of a number of operations to be performed in parallel are defined by the programmer, and their corresponding execution unit controls are generated at compile time and

loaded prior to program execution into a dedicated array in processor memory. Subsequently, the programmer invokes reference instructions to call these defined instructions, and passes parameters from regular instructions in program memory. As the regular instructions propogate down the processor's pipeline, they are replaced by the appropriate controls fetched from the dedicated array in processor memory, which then go directly to the execution unit for execution. These instructions may be redefined while the program is running. In this way the processor benefits from the speed of parallel processing without the chip area and power consumption overhead of a wide program memory bus and multiple instruction decoders. A simple syntax for defining instructions, similar to that of the C programming language is presented.

11 Claims, 8 Drawing figures

First Hit Fwd Refs

Generate Collection Print

L10: Entry 2 of 3 File: USPT Oct 15, 2002

US-PAT-NO: 6467036
DOCUMENT-IDENTIFIER: US 6467036 B1

TITLE: Methods and apparatus for dynamic very long instruction word sub-instruction selection for execution time parallelism in an indirect very long instruction word processor

DATE-ISSUED: October 15, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Pechanek; Gerald G.	Cary	NC		
Revilla; Juan Guillermo	Cary	NC		
Barry; Edwin F.	Cary	NC		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
BOPS, Inc.	Mountain View	CA			02

APPL-NO: 09/ 717992 [PALM]
DATE FILED: November 21, 2000

PARENT-CASE:
RELATED APPLICATIONS This is a continuation of application Ser. No. 09/205,588 filed on Dec. 4, 1998, now U.S. Pat. No. 6,173,389. The present invention claims the benefit of U.S. Provisional Application Ser. No. 60/067,511 entitled "Method and Apparatus For Dynamically Modifying Instructions in a Very Long Instruction Word Processor" and filed Dec. 4, 1997.

INT-CL: [07] G06 F 15/80

US-CL-ISSUED: 712/24; 711/220
US-CL-CURRENT: 712/24; 711/220

FIELD-OF-SEARCH: 712/24, 712/10, 712/20, 712/21, 712/22, 712/210, 712/200, 712/203, 712/208, 712/212, 712/215, 712/226, 711/220

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
6173389	January 2001	Pechanek et al.	712/24

☐

ART-UNIT: 2183

PRIMARY-EXAMINER: Treat; William M.

ATTY-AGENT-FIRM: Priest & Goldstein, PLLC

ABSTRACT:

A pipelined data processing unit includes an instruction sequencer and n functional units capable of executing n operations in parallel. The instruction sequencer includes a random access memory for storing very-long-instruction-words (VLIWs) used in operations involving the execution of two or more functional units in parallel. Each VLIW comprises a plurality of short-instruction-words (SIWs) where each SIW corresponds to a unique type of instruction associated with a unique functional unit. VLIWs are composed in the VLIW memory by loading and concatenating SIWs in each address, or entry. VLIWs are executed via the execute-VLIW (XV) instruction. The iVLIWs can be compressed at a VLIW memory address by use of a mask field contained within the XV1 instruction which specifies which functional units are enabled, or disabled, during the execution of the VLIW. The mask can be changed each time the XV1 instruction is executed, effectively modifying the VLIW every time it is executed. The VLIW memory (VIM) can be further partitioned into separate memories each associated with a function decode-and-execute unit. With a second execute VLIW instruction XV2, each functional unit's VIM can be independently addressed thereby removing duplicate SIWs within the functional unit's VIM. This provides a further optimization of the VLIW storage thereby allowing the use of smaller VLIW memories in cost sensitive applications.

24 Claims, 14 Drawing figures

First Hit Fwd Refs

End of Result Set

☐ Generate Collection Print

L12: Entry 1 of 1 File: USPT Sep 17, 2002

US-PAT-NO: 6453407
DOCUMENT-IDENTIFIER: US 6453407 B1

TITLE: Configurable long instruction word architecture and instruction set

DATE-ISSUED: September 17, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Lavi; Yoav	Raanana			IL
Rom; Amnon	Azor			IL
Knuth; Robert	Munich			DE
Blum; Rivka	Azor			IL
Yanni; Meny	Azor			IL
Granot; Haim	Azor			IL
Hershko; Anat	Azor			IL
Shenderovitch; Georgiy	Azor			IL
Cohen; Elliot	Raanana			IL
Weingatren; Eran	Tel Hashomer			IL

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Infineon Technologies AG	Munich			DE	03

APPL-NO: 09/ 247686 [PALM]
DATE FILED: February 10, 1999

INT-CL: [07] G06 F 9/22

US-CL-ISSUED: 712/24; 717/5
US-CL-CURRENT: 712/24

FIELD-OF-SEARCH: 712/24, 712/37, 717/4, 717/5

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL Clear

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	4435775	March 1984	Brantingham	711/118

☐	4607332	August 1986	Goldberg	714/8
☐	4610000	September 1986	Lee	365/189
☐	4897813	January 1990	Kumbasar	364/49
☐	4931989	June 1990	Rhodes	712/211
☐	5163139	November 1992	Haigh et al.	712/206
☐	5398321	March 1995	Jeremiah	712/216
☐	5450556	September 1995	Slavenburg et al.	712/235
☐	5634025	May 1997	Breternitz	712/207
☐	5748979	May 1998	Trimberger	712/37
☐	5774737	June 1998	Nakano	712/24
☐	5828897	October 1998	Kirsch	712/43
☐	5859993	January 1999	Snyder	712/208
☐	5950012	September 1999	Shiell	712/209
☐	5966534	October 1999	Cooke	717/5
☐	5983334	November 1999	Coon	712/23
☐	6105109	August 2000	Krumm	711/122
☐	6321322	November 2001	Pechanek	712/24

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
0 723 220	July 1996	EP	

OTHER PUBLICATIONS

"Selecting Predecoded Instructions with a Surrogate", IBM Technical Disclosure Bulletin, XP 000370750.
Joseph A. Fisher: "Very Long Instruction Word Architectures And The ELI-512", Yale University, New Haven, CT, 1983, pp. 140-151.
International Publication WO 97/50030 (Bauer et al.), dated Dec. 31, 1997.
J.A. Barber et al.: "MLID Addressing", IBM Technical Disclosure Bulletin, XP-002069681.

ART-UNIT: 2783

PRIMARY-EXAMINER: Coleman; Eric

ATTY-AGENT-FIRM: Greenberg; Laurence A. Stemer; Werner H. Mayback; Gregory L.

ABSTRACT:

A method for executing instructions in a data processor and improvements to data processor design, which combine the advantages of regular processor architecture and Very Long Instruction Word architecture to increase execution speed and ease of programming, while reducing power consumption. Instructions each consisting of a number of operations to be performed in parallel are defined by the programmer, and their corresponding execution unit controls are generated at compile time and

loaded prior to program execution into a dedicated array in processor memory. Subsequently, the programmer invokes reference instructions to call these defined instructions, and passes parameters from regular instructions in program memory. As the regular instructions propogate down the processor's pipeline, they are replaced by the appropriate controls fetched from the dedicated array in processor memory, which then go directly to the execution unit for execution. These instructions may be redefined while the program is running. In this way the processor benefits from the speed of parallel processing without the chip area and power consumption overhead of a wide program memory bus and multiple instruction decoders. A simple syntax for defining instructions, similar to that of the C programming language is presented.

11 Claims, 8 Drawing figures

First Hit

End of Result Set

Generate Collection Print

L11: Entry 2 of 2 File: DWPI Aug 26, 1999

DERWENT-ACC-NO: 1999-508926
DERWENT-WEEK: 200364

TITLE: Appts. for executing program instructions - with first instruction decoder discriminating if appts. is to execute referential instruction that initiates execution of instruction of different type

INVENTOR: AMNON, R; BLUM, R ; COHEN, E ; GRANOT, H ; HERSHKO, A ; KNUTH, R ; LAVI, Y ; SHENDEROVITCH, G ; WEINGARTEN, E ; YANNI, M ; ROM, A

PATENT-ASSIGNEE: INFINEON TECHNOLOGIES AG (INFN), SIEMENS AG (SIEI)

PRIORITY-DATA: 1998EP-0102925 (February 19, 1998)

Search Selected Search ALL Clear

PATENT-FAMILY:

	PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
☐	WO 9942922 A1	August 26, 1999	G	027	G06F009/318
☐	EP 942359 A1	September 15, 1999	E	000	
☐	CN 1291306 A	April 11, 2001		000	G06F009/318
☐	JP 2003525476 W	August 26, 2003		031	G06F009/30

DESIGNATED-STATES: CN IL JP US AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE AL AT BE CH DE DK ES FI FR GB GR IE IT LI LT LU LV MC MK NL PT RO SE SI

APPLICATION-DATA:

PUB-NO	APPL-DATE	APPL-NO	DESCRIPTOR
WO 9942922A1	February 4, 1999	1999WO-EP00849	
EP 942359A1	February 19, 1998	1998EP-0102925	
CN 1291306A	February 4, 1999	1999CN-0803152	
JP2003525476W	February 4, 1999	1999WO-EP00849	
JP2003525476W	February 4, 1999	2000JP-0532793	
JP2003525476W		WO 9942922	Based on

INT-CL (IPC): G06 F 9/30; G06 F 9/318; G06 F 9/38

ABSTRACTED-PUB-NO: WO 9942922A
BASIC-ABSTRACT:

The appts. has a first instruction decoder (1) which sequentially fetches program

instructions from a first program memory (2) to decode instructions of a first type. An address decoder (4) determines the address of data to load from or write to a data memory (3). According to the interpretation of the first instruction decoder, computational units (61-64) process the data and provide results. An execution logic unit (7) provides data for the units and controls the units operation according to the first type instructions.

The first instruction decoder discriminates if the appts is to execute a referential instruction that initiates execution of a second type of instruction. A second instruction decoder (9) fetches the second type of instruction and decodes it. The instruction has data assignment information of operands and of results. The referential instruction has address information of data on which the second type of instruction is to be executed. The appts. allows pipe-lined execution of all the instructions. The second type of instructions are stored in a second program memory (10).

ADVANTAGE - Flexible program memory organisation so lower program memory demand, maintains processor capability of parallel execution of instructions.

ABSTRACTED-PUB-NO: WO 9942922A
EQUIVALENT-ABSTRACTS:

CHOSEN-DRAWING: Dwg.1/3

DERWENT-CLASS: T01
EPI-CODES: T01-F03; T01-F03B; T01-F04;